PLS

OpenABEL Synthesis

Table of Contents

OpenABEL Synthesis

1. OpenABEL Format Syntax _____ 1
2. Information comment fields _____ 2
3. Identifier Construction (dot extension) _____ 2
4. .type specification _____ 3
5. Example 1: Simple combinatorial design _____ 4
6. Example 2: Sequential design _____ 4
7. Warning and error messages _____ 5
8. Remarks _____ 6
9. OpenABEL synthesis using PLS _____ 7

OpenABEL Synthesis

The OpenABEL, Truth-Table or PLA (TT) formats can be used to describe the behavior of combinatorial and sequential circuits.

1. OpenABEL Format Syntax

An example of the syntax of the OpenABEL format is shown in figure 1. Note that comments start with a # and end at the end of the line.

```
{#$ {PINS|NODES} <Integer> {<Information>}
                # optional information comment fields
{#$ <OtherKeywords> {<FieldInformation>}
                # Unsupported information comment fields
.i <integer>     # number of Inputs.
.o <integer>     # number of Outputs.
{.ilb {<Identifier>} +}
        # optional Input name list - allowed separators are Space, Tab or CR
        # if defined, size of the Input name list must be equal to the value of the .i
field.
{.ob {<Identifier>} +}
        # optional Output name list - allowed separators are Space, Tab or CR
        # if defined, size of the Output name list must be equal to the value
        # of the .o field.
{.p <integer> }
        # optional number of Product Terms (see below) in the file.
{.phase <Phase Vector> }
        # optional section defining whether Output are complemented or not
{.type <TypeValue> }
        # optional section defining the meaning of the digits in the
        # <Output Vector> field
{<Input Vector> <Output Vector>}*
        # Output definition
        # the number of "<Input Vector> <Output Vector>" lines must be equal
        # to the value of the .p field.
        # <Input Vector> and <Output Vector> are defined by:
<Input Vector> ::= any string made up of <value of .i> 0,1 or -, also called
                        "Product Term"
<Output Vector> ::= any string made up of <value of .o> 0,1 ,- or ~
<Identifier> ::= any char string, can be followed by a {+|-} to say whether the
                    signal is complemented or not
<OtherKeywords> ::= "TOOL" | "TITLE" | "MODULE" | "JEDECFILE" |
"DEVICE" |
                                "VECTORFILE" | "PROPERTY"
<Phase Vector> ::= any string made up of 0 (when the Output is complemented)
                                or 1 (when the Output is not  complemented)
<TypeValue> ::= "f" | "r" | "fd" | "fr" | "dr" | "fdr" | <StringValue>
<StringValue> is a string of length equal to the number of Outputs
                containing only "0", "1", "2", "3", "4" or "5"
                For example, assume 6 Outputs, then .type 121312 is a valid
                specification
```

Figure 1: *OpenABEL format syntax*

2. Information comment fields

The information comment fields are supported for the keywords PINS and NODES and are ignored for the other keywords.

The syntax for information comment fields is shown in figure 2.

```
{#$ {PINS|NODES} <Integer> {<Information>}
<Information> ::= {<PinOrNodeName> {:<PinOrNodeAssignement>}}+
<PinOrNodeName> ::= <Identifier>
<PinOrNodeAssignement> ::= <Integer>
```

Figure 2: *OpenABEL syntax for PINS and NODES*

3. Identifier Construction (dot extension)

The following dot extensions are supported for the definition of signal names:

```
.AP : Flip-flop asynchronous preset
.AR : Flip-flop asynchronous reset
.CE : Flip-flop clock enable
.CLK or .C : Clock Input
.CP : Flip-flop clock from pin
.D  : D flip-flop Input
.FB : Register feedback
.J  : J Input of a JK flip-flop
.K  : K Input of a JK flip-flop
.LE : Flip-flop latch enable (transparent low)
.LH : Flip-flop latch enable (transparent high)
.OE : Output enable control
.PIN : Output signal on a device pin
.PR : Flip-flop preset-synchronous or asynchronous
.Q  : Register feedback
.R  : R Input of a RS flip-flop
.RE : Flip-flop reset-synchronous or asynchronous
.REG : D flip-flop Input
.S  : S Input of a RS flip-flop
.SP : Flip-flop synchronous preset
.SR : Flip-flop synchronous reset
.T  : T flip-flop Input
.X1 and .X2 : The 2 entries of a XOR
```

Figure 3: *Supported dot extensions in OpenABEL syntax*

Remarks:

- The flip-flop synchronous preset and reset are used as asynchronous preset and reset

- The following dot extension are not supported : .FC and .LD

4. .type specification

The .type section is used to define the symbols used in the PLA matrix. The values f,r,fd,fr,rd and fdr correspond to the classical Espresso definition. Defining the .type through a string allows to give a different interpretation to the symbols in the PLA matrix for each Output.

Value '0' corresponds to f

Value '1' corresponds to r

Value '2' corresponds to fd (default value)

Value '3' corresponds to fr

Value '4' corresponds to dr

Value '5' corresponds to fdr

For example, assuming 4 Outputs, the following notations are equivalent :

.type fdr

.type 5555

With logical type f, for each Output,

- a '1' in the PLA matrix means that this product term belongs to the ON-set

- a '0', a '~' or a '-' means that this product term has no meaning for the value of the function.

This logical type corresponds to an actual PLA where only the ON-set is actually implemented.

With logical type r, for each Output,

- a '0' in the PLA matrix means that this product term belongs to the OFF-set,

- a '1', a '~' or '-' means that this product term has no meaning for the value of the function.

With logical type fd (default), for each Output,

- a '1' in the PLA matrix means that this product term belongs to the ON-set,

- a '0' or a '~' means that this product term has no meaning for the value of the function,

a '-' means that this product term belongs to the DC-set.

With logical type dr, for each Output,

- a '0' in the PLA matrix means that this product term belongs to the OFF-set,

- a '-' means that this product term belongs to the DC-set.

- a '1' or a '~' means that this product term has no meaning for the value of the function,,

With logical type fr, for each Output,

- a '1' in the PLA matrix means that this product term belongs to the ON-set,

- a '0' means that this product term belongs to the OFF-set

- a '-' or a '~' means that this product term has no meaning for the value of the function.

With logical type fdr, for each Output,

- a '1' in the PLA matrix means that this product term belongs to the ON-set,

- a '0' means that this product term belongs to the OFF-set

- a '-' means that this product belongs to the DC-set

- a '~' means that this product term has no meaning for the value of the function.

5. Example 1: Simple combinatorial design

The following lines give the specification of a 5-inputs / 3-outputs combinatorial design

```
.i 5                    #number of Inputs
.o 3                    #number of Outputs
.ilb a b c d e          #names of Inputs (optional)
.ob f1 f2 f3            #names of Outputs (optional)
.p 4                    # Number of product terms in the file
1--0-   111             # First product term :
                        # a*!d appears in functions f1 f2 f3 ( ! is negation)
111--  011
0----           -10
00000           -10
.e                      # end of file
```

Figure 4: *Simple combinatorial design using OpenABEL syntax*

6. Example 2: Sequential design

In this design the D Input of the flip-flop is connected to ((A.B) xor (C.D)), the Q Output of the flip-flop is connected to an inverting 3-state gate with low polarity OE.

```
#$ TOOL ABEL 4.00
#$ MODULE xor
#$ JEDECFILE u2
#$ DEVICE P20X10
#$ PINS 7 Clk:1 OE:14 A:2 B:3 C:4 D:5 Q16-:16
.i 7
.o 4
.ilb Clk OE A B C D
.ob Q16.D.X1 Q16.D.X2 Q16.CLK Q16.OE
.phase 1111
.p 4
--11-- 1000
----11 0100
1----- 0010
-0---- 0001
.e
```

Figure 5: *Sequential design using OpenABEL syntax*

7. Warning and error messages

The following warnings/errors may be displayed when reading a Truth-Table format file :

-Warning PRESET "PresetSignalName" will be used as asynchronous PRESET

-Warning RESET "ResetSignalName" will be used as asynchronous RESET

Synchronous reset and preset are used as asynchronous reset and preset

-"DotExtension" extension unknown

The dot extension found is unknown and will be ignored.

-The Gate "GateName" has no Output

-The Gate "GateName" has no Input

-XOR "XORName" with bad number of entry not created

For example if the Q16.D.X2 is missing, the following message will be displayed :

XOR Q16.D with bad number of entry not created

8. Remarks

It is checked that the given ON, OFF and DC sets are disjoint, if not, one of the following warning/error messages is printed according to the ".type" value :

"Warning: ON-SET and DC-SET intersect"

"Warning: OFF-SET and DC-SET intersect"

"Error: ON-SET and OFF-SET intersect"

"Error: OFF-SET and DC-SET intersect"

"Error: ON-SET and DC-SET intersect"

The example of figure 6 will cause a warning because "00" belongs to the DC and ON sets.

```
.i 2
.o 1

00        1
0-        -
.e
```

Figure 6: *ABEL disjoint DC and ON sets create a warning*

Beware that when "ON-SET and DC-SET intersect", some points of the ON set may be moved to the OFF set during minimization.

For example, given the following specification :

f = {ON} a.!b + a.c ;

f = {DC} a.b.c ;

the result after minimization will be : f = {ON} a.!b ;

The same result is obtained when "OFF-SET and DC-SET intersect" : some points of the OFF set may be moved to the ON set during minimization.

For example, given the following specification (.type dr) :

f = {OFF} a + b ;

f = {DC} a.!b ;

the result after minimization will be : f = {ON} !b ;

Note that these specifications will result in an error if the specified ".type" is changed to fdr.

9. OpenABEL synthesis using PLS

From the command menu of the Graphical User Interface (GUI), select the Execute menu. Then select the OpenABEL format from the list of available inputs (Input Format) as shown in figure 7.

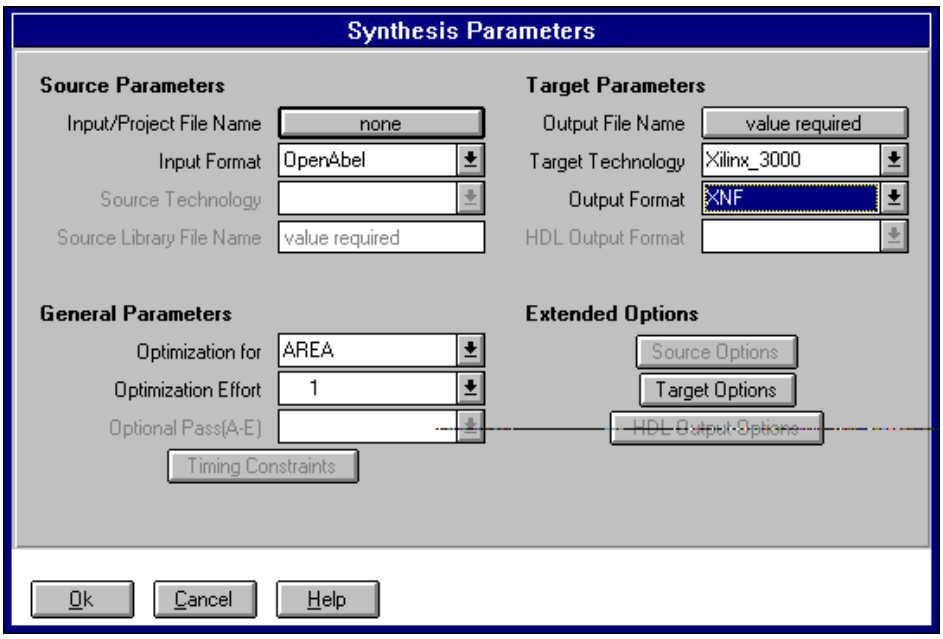

Figure 7: *Selecting OpenABEL input format in PLS menus.*

Then using the selection boxes, select all other appropriate synthesis parameters. Refer to the technology pages of the user manual for details on the technology specific options.